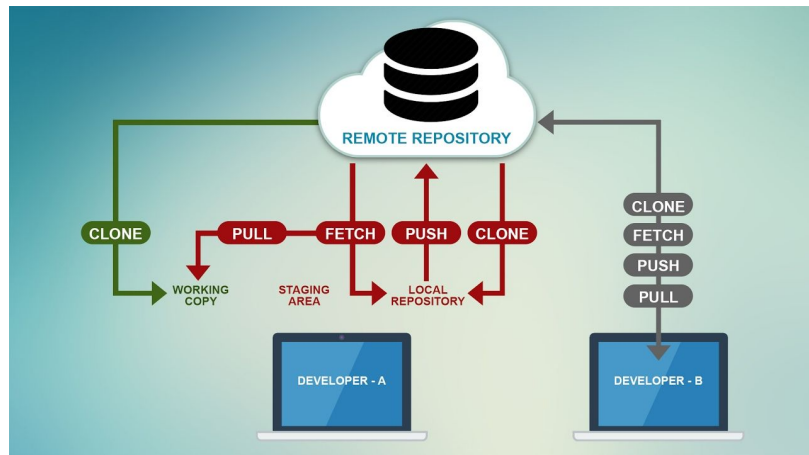




Cometer errores es humano, pero para estropear realmente las cosas necesitas un ordenador

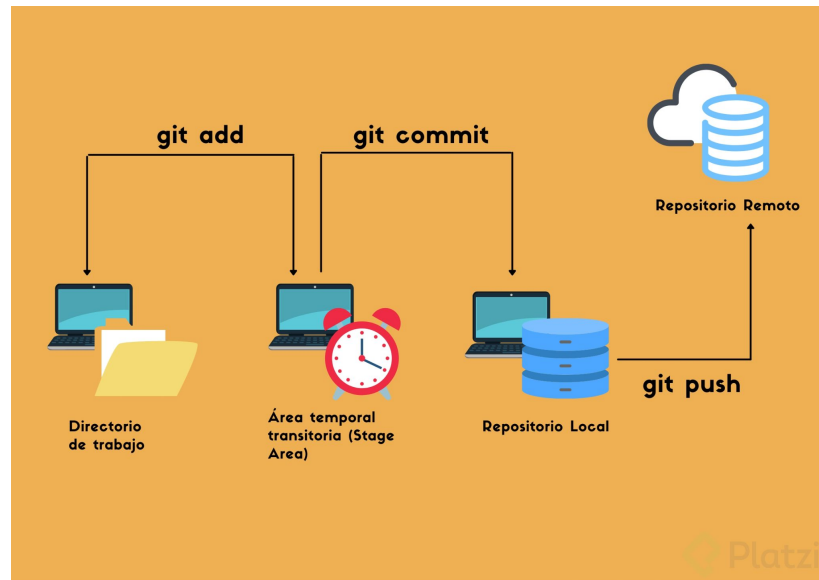
Git - local remoto

- Cuando se trabaja con Git cada desarrollador tiene una copia del repositorio en local.
- Push
 - El repositorio en remoto es el que manda, y no se actualiza hasta que no hacemos un push.
- Pull
 - Para actualizar nuestro repositorio tenemos que hacer un pull.
- Fetch
 - Sirve para que en nuestro repositorio en local se actualicen las ramas que no tenemos.



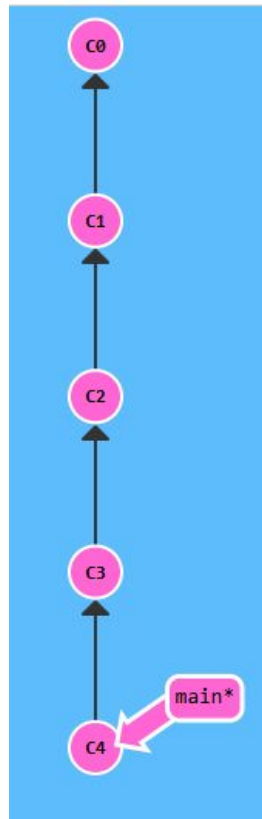
Git - Local

- Cuando estamos trabajando en local, podemos hacer un número de commits para actualizar el repo.
- Estos commits solo tendrán efecto en nuestro repositorio local, hasta que no subamos los cambios.
- Podemos hacer tantos commits como queramos.



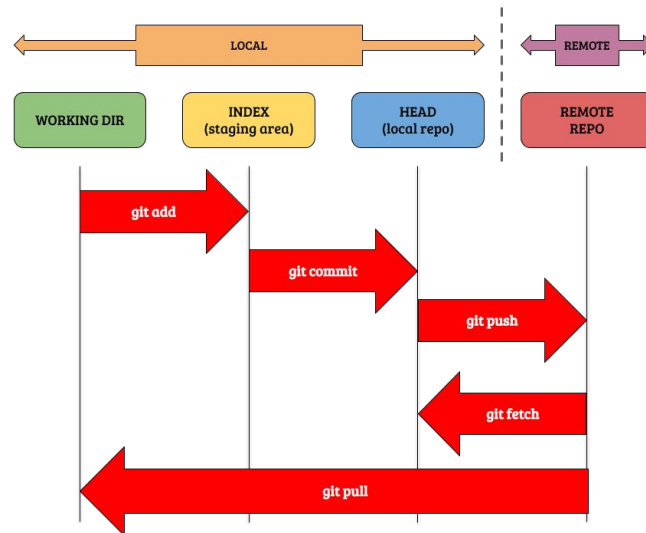
Git - commit

- Un commit es un conjunto de ficheros cambiados en nuestro repositorio.
- Se suceden uno detrás de otro, apuntando al commit anterior para saber su referencia.
- Cada commit tiene un mensaje y un id hasheado.
- Además del conjunto de ficheros que han cambiado, han sido borrados o han sido creados.

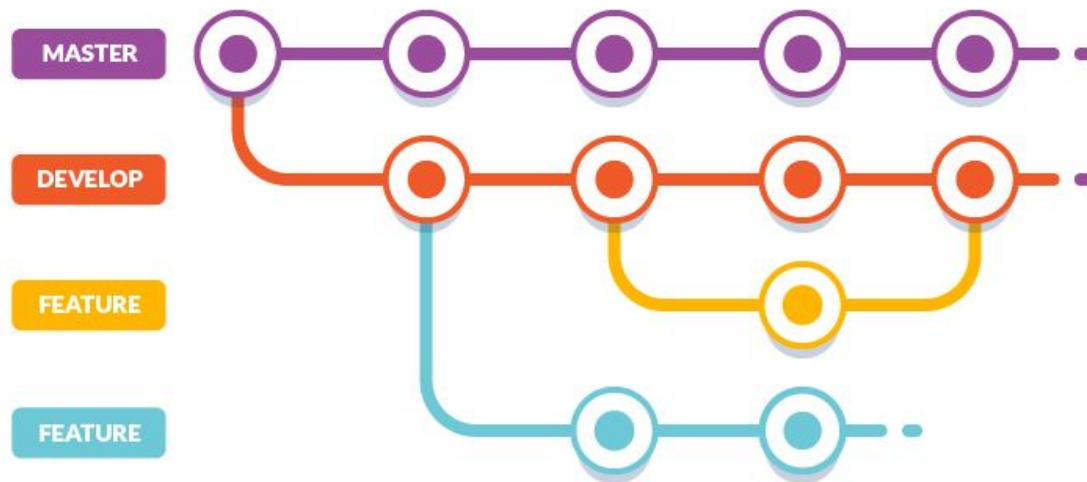


Git - ¿Cómo hacer un commit?

- Para hacer un commit hay que seleccionar los ficheros que queremos añadir a ese commit.
- Para ello nos valemos del comando *git add* que selecciona los ficheros que queremos poner en el commit.
- Con *git status* podemos ver el estado de los ficheros de nuestro directorio, y ver cuál está para añadirse al commit y cual no.
- El commit se haría en local, y hasta que no hagamos push no se subiría al repo remoto.

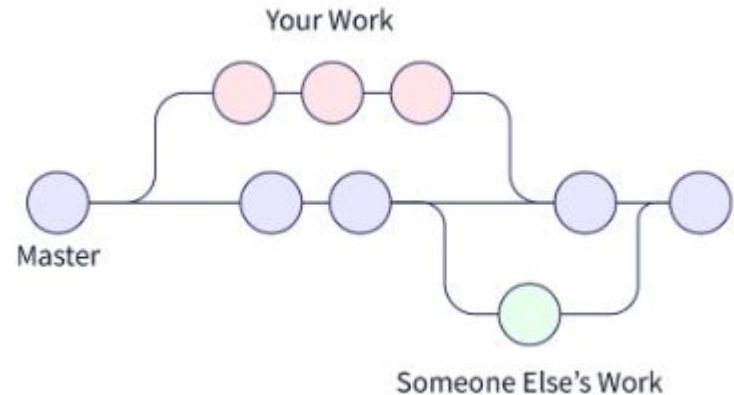


Git - ¿Qué es una rama?



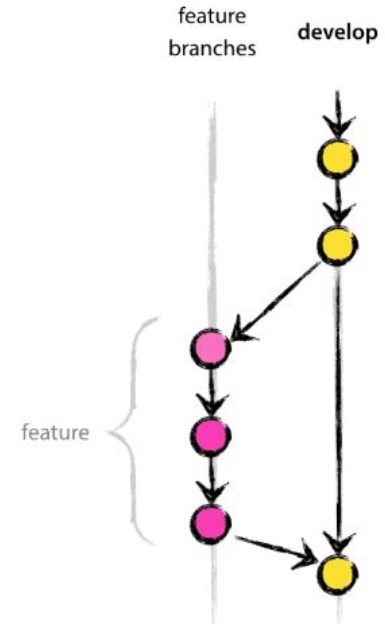
Git - Ramas

- Cada desarrollador saca una rama de la rama principal
 - Normalmente llamada rama *main*.
- Desde esta rama hace sus cambios para crear una Pull Request.
- Con la Pull Requests, los cambios del desarrollador se integran en la rama principal.
- En las PRs puede haber conflictos que tenemos que resolver.



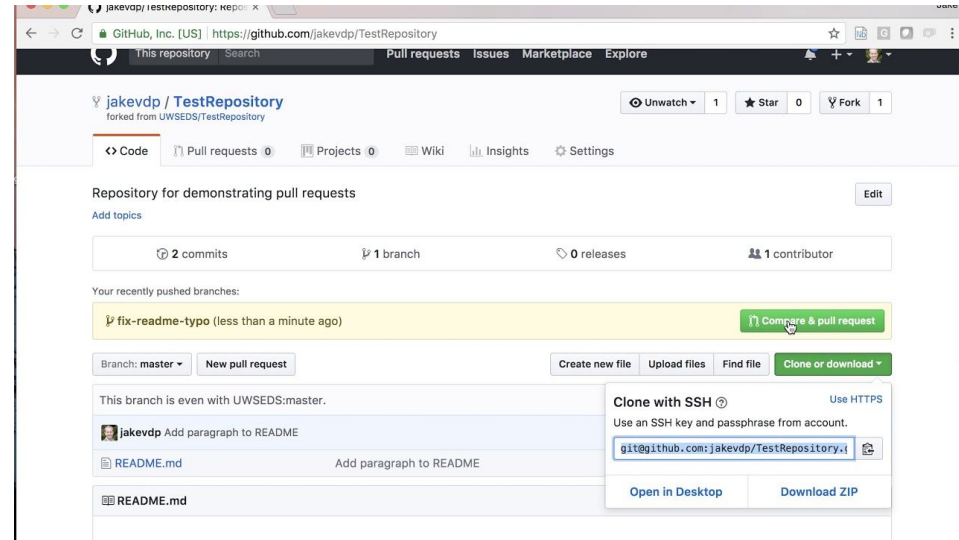
Git - Ramas

- Para crear una rama se puede usar el comando `git checkout -b <nombre-rama>`
- Para navegar entre ramas podemos usar el comando `git checkout <nombre-rama>`
- Para subir los cambios de una rama podemos usar el comando `git push origin <nombre-rama>`
- Para bajarte los cambios de una rama se puede usar el comando `git pull origin <nombre-rama>`



Git - Pull Requests

- Las Pull Requests se crean en el repositorio remoto
- Tenemos que tener la rama subida.
- Con la rama subida, creamos una PR seleccionando la rama origen y la rama destino
 - Normalmente la rama destino será *main*
- Una vez creada la PR, la podremos revisar y cuando esté todo claro mergearla en el repo.



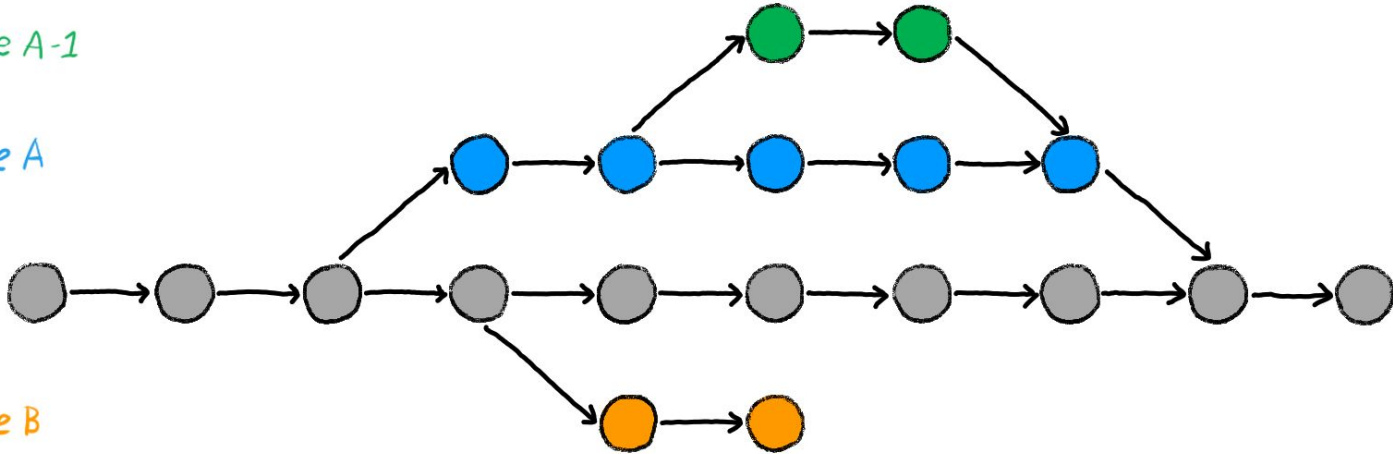
Git - Ramas

Feature A-1

Feature A

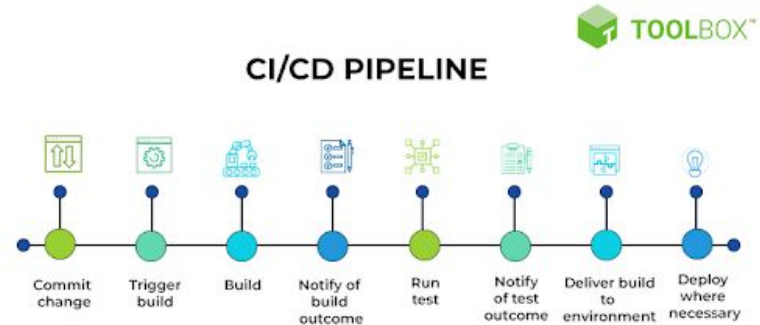
Main

Feature B

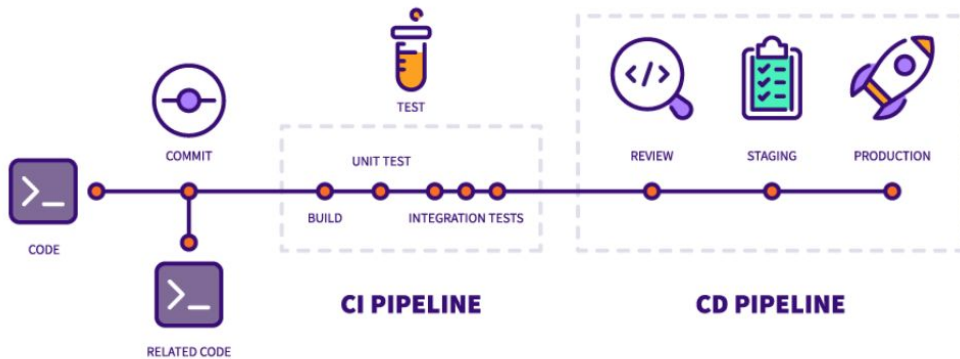


Git - Pull Requests

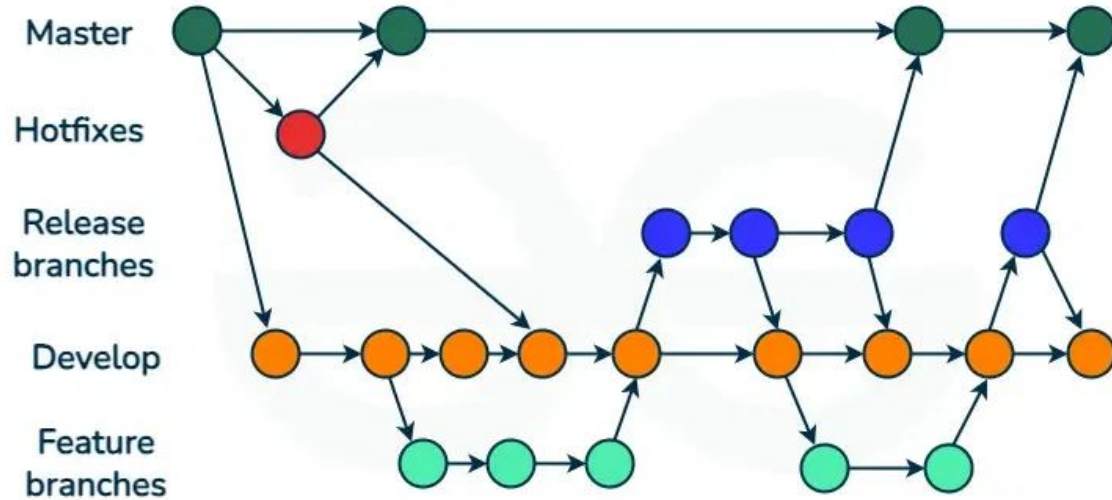
- Cuando se mergea una Pull Requests, normalmente saltan los mecanismos de CI y CD.
- La integración continua verificará que no se ha roto nada de la aplicación. (CI)
- El despliegue continuo desplegará la aplicación donde corresponda. (CD)



Git - CI CD



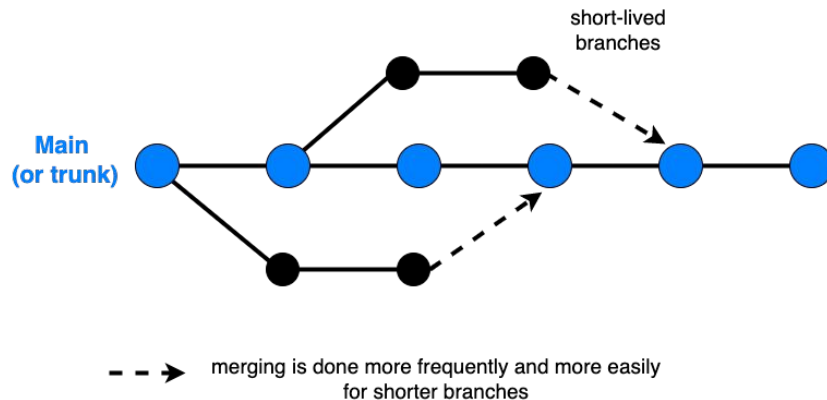
Git - Master/develop estrategia



Git - Trunk based development

Trunk-based development

StatusNeo





Git - chuleta subir cambio

```
git pull origin main
```

```
git checkout -b feature/button-component
```

```
git status
```

```
git add . => . es la carpeta raíz
```

```
git commit -m "<tipo cambio>: <mensaje>" => git commit -m "feat: add button component"
```

```
git push origin feature/button-component
```

Create Pull Request in Github.



Git - solucionar conflictos

`git pull origin main` => Hay conflictos.

`git checkout -b feature/button-component`

`git status`

`git add .` => . es la carpeta raíz

`git commit -m "<tipo cambio>: <mensaje>"` => `git commit -m "feat: add button component"`

`git checkout main`

`git pull origin main`

`git checkout feature/button-component`

`git merge main` => Solucionar conflictos en VSCode. => commit del merge.

`git push origin feature/button-component`

Create Pull Request in Github.